**SHOPPING CART PROJECT**

**Submitted By:** GROUP NO 8

**Roll No:** 233536 , 233532 , 233590

**Course Title:** VISUAL PROGRAMMING

**Semester:** 3RD

**Session:** 2023-2027

**Department:** COMPUTER SCIENCE

**Submitted To:** MAM AATKA ALI



**AIR UNIVERSITY MULTAN**

Table of Contents

[**1. Explanation of Classes 3**](#_Toc180776593)

[**1.1 User Class 3**](#_Toc180776594)

[**1.2 Product Class 3**](#_Toc180776595)

[**1.3 Shop Class 3**](#_Toc180776596)

[**1.4 ShoppingCart Class 4**](#_Toc180776597)

[**1.5 UML Diagram 4**](#_Toc180776598)

[**2. Explanation of Lists 5**](#_Toc180776599)

[**3. Categories and Stored Items 5**](#_Toc180776600)

[**4. User Login and File Storage 6**](#_Toc180776601)

[**5. Conclusion 6**](#_Toc180776602)

[**6. Detailed Explanation of Categories and Lists 6**](#_Toc180776603)

[**7. User Interaction: Adding, Removing, and Viewing Items in the Cart 7**](#_Toc180776604)

[**7.1 Adding Items to the Cart 7**](#_Toc180776605)

[**7.2 Removing Items from the Cart 7**](#_Toc180776606)

[**7.3 Viewing the Cart 7**](#_Toc180776607)

[**8. User Flow: Signup, Login, and Checkout Process 8**](#_Toc180776608)

[**9. Future Enhancements 8**](#_Toc180776609)

Shopping Cart Program Explanation

This document provides a detailed explanation of the shopping cart program, highlighting the main components such as classes, lists, user creation, login, file storage, product categories, and cart management.

# 1. Explanation of Classes

## 1.1 User Class

The `User` class represents a user in the program with attributes like name, email, password, phone number, and age. It has a constructor for initializing user attributes and an override of the `ToString` method to convert the user data into a comma-separated string format for file storage.

## 1.2 Product Class

The `Product` class represents a product with attributes like name, price, discount, and available quantity. It includes a method `GetDiscountedPrice` to calculate the price after applying the discount. Products are categorized into various categories like Electronics, Clothing, Footwear, Home Appliances, etc.

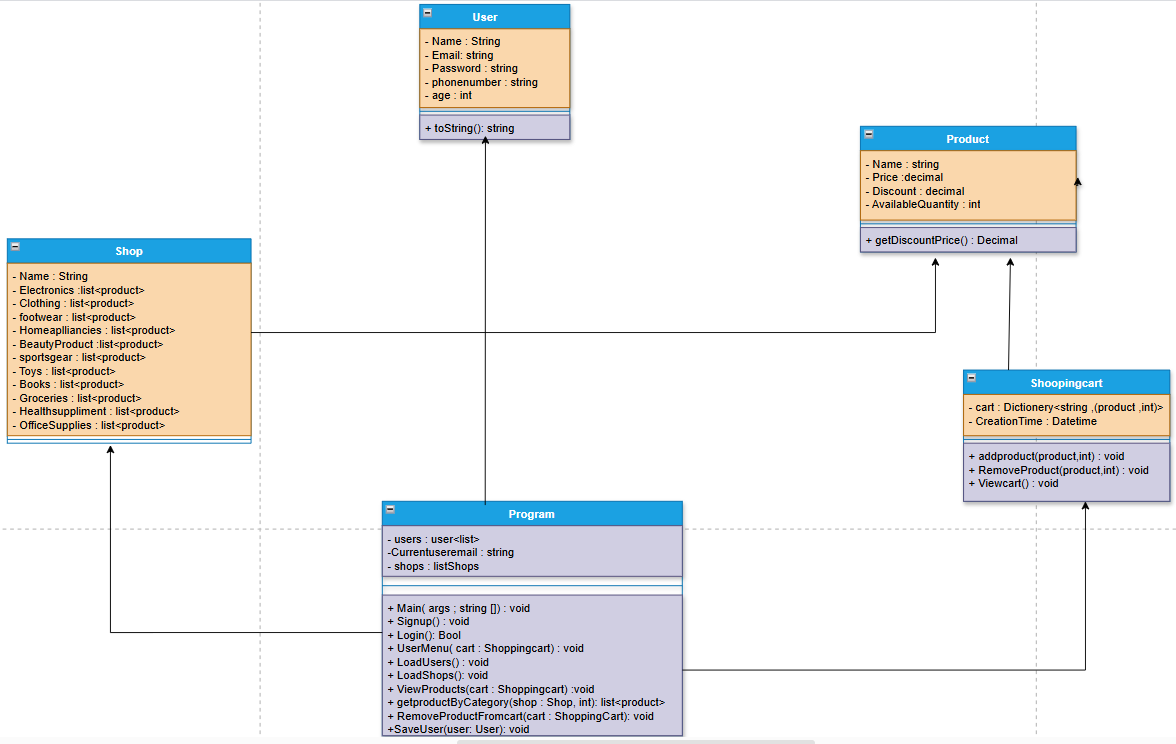
## 1.3 Shop Class

The `Shop` class represents a shop with a name and multiple lists of products in various categories like electronics, clothing, footwear, etc. It initializes empty lists for each category to store products and allows loading sample data for each shop.

## 1.4 ShoppingCart Class

The `ShoppingCart` class manages the shopping cart functionality, such as adding products, removing products, viewing the cart, and calculating total costs with sales tax. It uses a dictionary to store the product and quantity added to the cart. The class also handles cart expiration by checking if the cart has been active for more than 30 minutes.

## 1.5 UML Diagram



# 2. Explanation of Lists

Lists are used in various places in the program to store users, shops, and products. For instance:

**1.** `users` stores the list of registered users, which is populated during signup and loaded from a file.

**2.** `shops` is a list of different shop objects, each containing multiple categories of products.

**3.** Each shop contains lists for different product categories like Electronics, Clothing, etc., which store individual product objects.

# 3. Categories and Stored Items

The program defines several categories in the `Shop` class, such as:

**1.** Electronics

**2.** Clothing

**3.** Footwear

**4.** Home Appliances

**5.** Beauty Products

**6.** Sports Gear

**7.** Toys

**8.** Books

**9.** Groceries

**10.** Health Supplements

**11.** Office Supplies

Each category stores products with attributes like name, price, discount, and available quantity. For example, products like T-shirts, sneakers, smartwatches, and perfumes are stored in these categories.

# 4. User Login and File Storage

The program allows users to sign up and log in. When signing up, the user provides their name, email, password, phone number, and age. This data is then stored in a text file (`users.txt`) in a comma-separated format. The `LoadUsers` method reads this file to populate the `users` list, and new users are appended to the file using the `SaveUser` method. During login, the user’s email and password are validated against the stored data.

# 5. Conclusion

In conclusion, this shopping cart program is designed to simulate a basic e-commerce platform. It allows users to sign up, log in, add products from various categories to a shopping cart, remove items, and view the total cost after applying discounts and sales tax. The program also handles cart expiration and stores user data in a text file for persistence.

# 6. Detailed Explanation of Categories and Lists

Categories are implemented as lists within the `Shop` class. Each category, such as Electronics, Clothing, Footwear, and others, stores products relevant to that category. Each product is an instance of the `Product` class and has attributes like name, price, discount, and available quantity. The list structure allows for easy manipulation, like adding or removing products, as well as iterating through the list to display available products to the user.

Each shop has its own set of lists representing different product categories. This structure allows the program to support multiple shops, each with a unique selection of products. For example, the program includes a "Puma" shop with items like smartwatches, clothing, and footwear, and an "Adidas" shop with similar but distinct products.

# 7. User Interaction: Adding, Removing, and Viewing Items in the Cart

The `ShoppingCart` class allows users to interact with the shopping cart by adding, removing, and viewing products. Here’s how each interaction is handled:

### 7.1 Adding Items to the Cart

Users can add products to the cart by selecting the product category from a shop and choosing a product. The program first checks if the requested quantity is available in stock. If sufficient quantity is available, the program adds the product to the cart and deducts the quantity from the available stock.

## 7.2 Removing Items from the Cart

The user can also remove items from the cart. When removing an item, the program checks if the quantity to remove is available in the cart. If the requested quantity is greater than what’s in the cart, the program alerts the user. The program updates both the cart and the product’s available stock accordingly.

## 7.3 Viewing the Cart

At any time, users can view the contents of their cart. The program displays each product’s name, quantity, discounted price, and the total cost of items in the cart, including sales tax. The sales tax is calculated at a rate of 10%. If the cart has expired (after 30 minutes), users are notified that their cart is no longer valid, and they must create a new cart.

# 8. User Flow: Signup, Login, and Checkout Process

**1. Signup:** New users must provide their name, email, password, phone number, and age to create an account. This data is stored in a file (`users.txt`) for persistence across sessions.

**2. Login:** Existing users log in by entering their email and password. The program verifies these details against the stored data. If credentials are correct, the user can access the shopping features.

**3. Checkout Process:** Although this version does not include an explicit checkout process, the cart’s total calculation feature gives users a clear idea of how much they would need to pay, including discounts and taxes.

# 9. Future Enhancements

This program could be extended by adding a proper checkout process, integrating payment gateways, or adding features like product reviews and user profiles. Additionally, improving error handling and implementing product filtering by price or rating would enhance the user experience.